In the Claims

Please cancel claims 1-4, and 11, and amend claims 6, 12, 15, 20, 21, 23-25, 27 and 28 as follows:

1. (cancelled) ~~A certificate for Public Key Infrastructure (PKI) wherein the certificate validity is determined by the amount of ciphertext associated with the certificate.~~

2. (cancelled) ~~A certificate according to claim 1 wherein when the amount of ciphertext generated is below a predetermined value, the certificate is valid, and when the amount of ciphertext generated reaches a predetermined value, the certificate is invalid.~~

3. (cancelled) ~~A certificate according to claim 2 wherein the certificate validity is also dependent on the elapsed time and revocation status.~~

4. (cancelled) ~~A certificate for a PKI system according to claim 2, wherein the certificate validity is defined by~~

$$\text{(Certificate\_Validity)} = \frac{k}{(Ciphertext\_Generated) + (Elapsed\_Time)} \wedge (\text{Revocation\_Status})$$

~~where k is a constant value representing the assurance level of the keys in use.~~

5. (original) A certificate for a PKI system according to claim 4 compatible with the X.509 standard.

6. (currently amended) A certificate ~~according to claim 4~~ for Public Key Infrastructure (PKI), the certificate validity being determined by the amount of ciphertext associated with the certificate,
wherein when the amount of ciphertext generated is below a predetermined value, the certificate is valid, and when the amount of ciphertext generated reaches a predetermined value, the certificate is invalid,
comprising:

an extension including a Certificate Ciphertext Entitlement (CCE) value defining the amount of data that it is permissible for a certificate to encrypt before it must be rendered invalid;

an object identifier defining the units for ciphertext entitlement; and

an associated Ciphertext Generated Index (GCI) defining the count of how much cyphertext has been encrypted by the key,

the certificate validity also being dependent on the elapsed time and revocation status

wherein the certificate validity is defined by

$$(Certificate\_Validity) = \frac{k}{(Ciphertext\_Generated) + (Elapsed\_Time)} \wedge (Revocation\_Status)$$

where k is a constant value representing the assurance level of the keys in use.

7. (original) A certificate according to claim 6 wherein the extension also defines a version of the Ciphertext limited certificates in effect for the certificate.

8. (original) A certificate according to claim 6 wherein the CCE is expressed as a non-critical extension to a X.509 certificate.

9. (original) A certificate according to claim 6 wherein the CCE included in the signed body of the certificate.

10. (original) A certificate according to claim 8 wherein CCE default values are dependent on assurance level assigned to the certificate.

11. (cancelled) ~~A method of managing ciphertext devaluation in a PKI, comprising:~~

~~determining a certificate ciphertext entitlement (CCE);~~

~~calculating a generated ciphertext index (GCI) and~~

~~performing a certificate ciphertext entitlement threshold detection~~

~~and when the GCI reaches or exceeds the CCE, causing a key update.~~

12. (currently amended) A method according to claim ~~11~~ 15 wherein the key update is implemented as a rollover of the certificate or by invalidating the certificate.

13. (original) A method according to claim 12 wherein the key update is implemented as an immediate rollover.

14. (original) A method according to claim 12 wherein the key update is implemented at next log-in.

15. (currently amended) A method ~~according to claim 11,~~ of managing ciphertext devaluation in a PKI, comprising:

determining a certificate ciphertext entitlement (CCE)

calculating a generated ciphertext index(GCI)

wherein calculating the generated ciphertext index (GCI) comprises decrypting and verifying the decryption log

performing a certificate ciphertext entitlement threshold detection

and when the GCI reaches or exceeds the CCE, causing a key update.

16. (original) A method according to claim 15, comprising generating a time stamped decryption log.

17. (original) A method according to claim 15 comprising,

when data is decrypted, checking for a unique identifier associated with each ciphertext archive that has been decrypted,

and if the unique identifier is found, the GCI is not updated

and when the unique identifier is not found in the decryption log, updating the decryption log and adding the size of the current decrypted data to the GCI.

18. (original) A method according to claim 17 wherein the unique identifier is the hash of the symmetric key used to encrypt the data.

19. (original) A method according to claim 18 wherein the decryption log is kept only for ciphertext archives that have been encrypted using the most current key pair.

20. (currently amended) A method according to claim ~~11~~ 15 wherein the GCI is stored in bytes and the GCI is converted into units corresponding to the Certificate ciphertext Entitlement during threshold detection.

21. (currently amended) A method according to claim ~~11~~ 15 wherein the decryption log and GCI are signed and encrypted by the certificate subject.

22. (original) A method according to claim 15 wherein the GCI is contained in the decryption log.

23. (currently amended) A method according to claim ~~11~~ 15 wherein the step of performing a certificate ciphertext entitlement threshold detection is performed each time decryption takes place.

24. (currently amended) A method according to claim ~~11~~ 15 wherein the step of performing a certificate ciphertext entitlement threshold detection is performed at log in.

25. (currently amended) A method ~~according to claim 11~~ of managing ciphertext devaluation in a PKI, comprising:

determining a certificate ciphertext entitlement (CCE);

calculating a generated ciphertext index(GCI);

performing a certificate ciphertext entitlement threshold detection and

when the GCI reaches or exceeds the CCE, causing a key update,

wherein the step of performing a certificate ciphertext entitlement threshold detection comprises decrypting the GCI, verifying the digital signature, converting the GCI to ~~unites~~ units stipulated in the CCE extension, comparing the GCI to the CCE and if GCI is

greater than or equal to the CCE, requesting a key update in accordance with policy requirements.

26. (original) A method according to claim 25 wherein after the key update has taken place, clearing the existing decryption log and GCI to reset the count.

27. (currently amended) A system for managing ciphertext devaluation in a PKI, comprising:

means for determining a certificate ciphertext entitlement (CCE)

means for calculating a generated ciphertext index (GCI)

means for performing a certificate ciphertext entitlement threshold detection comprising means for decrypting the GCI, verifying the digital signature, converting the GCI to units stipulated in the CCE extension, and comparing the GCI to the CCE

and means for causing a key update when the GCI reaches or exceeds the CCE.

28. (currently amended) A computer readable medium for implementing a method of managing ciphertext devaluation in a PKI, comprising:

determining a certificate ciphertext entitlement (CCE)

calculating a generated ciphertext index(GCI) and

performing a certificate ciphertext entitlement threshold detection comprising decrypting the GCI, verifying the digital signature, converting the GCI to units stipulated in the CCE extension, and comparing theGCI to the CCE

and, when the GCI reaches or exceeds the CCE, causing a key update.